JUPITER PRIME

Consulting • Training • Advisory

جوبيتر برايم للاستشارات والتدريب

COURSE BROCHURE

Relational Database Design Tools and Techniques Training Course

4 – 8 January 2027
Barcelona

Relational Database Design Tools and Techniques Training Course

Ref: 1552_57165 **Date:** 4 – 8 January 2027 **Location:** Barcelona **Fees:** 6200 GBP

Relational Database Design Tools and Techniques is a 1-week training course for data analysts, developers and information managers who model business data and need to produce a validated relational schema for a real business case. Analytics dashboards, machine learning features and AI assistants all depend on tables whose keys, constraints and joins can be trusted, so schema choices now shape reporting accuracy and model quality. Jupiter Prime Training delivers this course for practitioners who own that foundation.

Why Relational Database Design Tools and Techniques Matters Now

Three pressures meet in the schema: analytics teams asking for faster, consistent joins; AI and machine learning teams needing clean, versioned feature tables; and auditors expecting traceable lineage from source record to report. A well-normalised design with clear keys cuts duplicate records, update anomalies and slow queries, while a poor one multiplies reconciliation effort and feeds models with inconsistent inputs. Where tables feed AI systems, the course links design choices to data quality expectations in ISO/IEC 42001, the AI management system standard, and the NIST AI Risk Management Framework NIST AI RMF.

Who Gains Most from Relational Database Design Tools and Techniques

- Analysts who write SQL against operational and warehouse tables and reconcile figures across reports
- Application developers who create and migrate tables behind business systems
- Information managers who own data dictionaries, retention rules and access rights
- Data engineers who load feature tables for machine learning and AI workloads
- Business intelligence specialists who build dimensional models for dashboards

Data functions gain most when analysts, developers and information owners attend together, because naming conventions, key rules and constraint choices only hold when every author of the schema applies them.

Relational Database Design Tools and Techniques Capabilities

Participants move from a business conversation to tables, keys and indexes they can defend line by line.

- Build entity-relationship diagrams from business rules and stakeholder interviews — sign-off by process owners
- Apply first to third normal form and Boyce-Codd checks — decisions on splitting or merging tables
- Specify primary, foreign and surrogate keys with constraints — data integrity owners
- Design indexes from query plans and workload patterns — performance trade-offs for report users
- Weigh normalised against dimensional and denormalised layouts — analytics and AI feature consumers
- Test a schema with sample data, constraint violations and migration scripts — release approval

Key Relational Database Design Tools and Techniques Terms

- Entity-relationship diagram — a visual model of entities, their attributes and the relationships and cardinalities that connect them.
- Normalisation — the stepwise restructuring of tables to remove redundancy and update anomalies, measured against normal forms.
- Primary key — a column or set of columns whose value uniquely identifies each row in a table.
- Foreign key — a column that references a primary key in another table, enforcing valid relationships between rows.
- Index — a stored structure, usually a B-tree, that speeds lookups and joins at the cost of extra writes and storage.
- Star schema — a dimensional layout with one fact table of measures surrounded by descriptive dimension tables for reporting.

Relational Database Design Tools and Techniques Day by Day

Day 1: Data Requirements and the Relational Model

- Relational Model Principles — Tables, Tuples and Set-Based Thinking
- Business Rule Elicitation — Turning Interviews into Modelling Statements
- Conceptual, Logical and Physical Models — Choosing the Right Abstraction
- Workload Profiling for Analytics and AI — Read, Write and Feature Patterns
- Data Dictionary and Naming Conventions — Shared Vocabulary Across Authors

Day 2: Entity-Relationship Modelling

- Entities, Attributes and Cardinality — Capturing One-to-Many and Many-to-Many Links
- Associative Entities and Junction Tables — Resolving Many-to-Many Relationships
- Subtypes and Supertypes — Modelling Customers, Suppliers and Employees
- ER Diagramming Tools — Forward Engineering Diagrams into DDL
- Model Walkthroughs with Process Owners — Validating Rules Before Build

Day 3: Normalisation, Keys and Constraints

- First to Third Normal Form — Removing Repeating Groups and Partial Dependencies
- Boyce-Codd and Functional Dependencies — Spotting Hidden Anomalies
- Natural Against Surrogate Keys — Stability, Privacy and Join Cost
- Check, Unique and Referential Constraints — Enforcing Integrity in the Engine
- Temporal and History Tables — Tracking Changes for Audit and Model Training

Day 4: Physical Design for Analytics and AI Workloads

- Index Types and Query Plans — Choosing B-Tree, Composite and Covering Indexes
- Partitioning and Data Types — Storage, Speed and Precision Choices
- Star Schema and Dimensional Modelling — Serving Dashboards and Ad Hoc SQL
- Feature Tables and Data Lineage — Feeding Machine Learning Reliably
- Views, Roles and Row-Level Security — Controlling Access to Sensitive Columns

Day 5: Practising the Validated Relational Schema

- Business Case ER Model — Applying Day 2 Modelling to a Live Scenario
- Normalisation Review of Draft Tables — Applying Day 3 Dependency Checks
- Index and Partition Plan — Applying Day 4 Query Plan Evidence
- DDL Scripts, Sample Data and Migration Tests — Proving the Design Runs
- Validated Relational Schema — Final Design Pack for Peer Approval

Applying Relational Database Design Tools and Techniques After the Course

- Run a normalisation and key review on one high-traffic reporting table set and record the anomalies found.
- Profile the ten slowest analytical queries with execution plans and propose index or partition changes for sign-off.

Relational Database Design Tools and Techniques Questions FAQ

What is relational database design, and why does it matter now?

Relational database design is the discipline of structuring data into related tables with keys and constraints. It matters now because analytics, reporting and AI models all inherit the quality of those tables, so design errors spread quickly into decisions.

Which roles gain most from relational database design tools and techniques?

Data analysts, application developers, information managers, data engineers and BI specialists gain most. Each writes or reads the schema daily, so shared modelling rules reduce rework, duplicate records and conflicting report figures.

How does relational database design differ from a general SQL querying course?

It focuses on structuring tables rather than writing queries against them. You decide entities, keys, normal forms and indexes, while a querying course assumes the schema exists and teaches retrieval and aggregation techniques.

What do you bring back from relational database design tools and techniques training?

You bring back a tested schema for a real business case, with diagrams, DDL scripts, constraint rules and an index plan. It gives your colleagues a reference design to review, extend and reuse on later projects.

Relational Database Design Tools and Techniques: The Way Forward

The validated relational schema first informs whether an existing system needs restructuring or only new constraints and indexes. Developers use it to script migrations, analysts to write consistent joins, and information managers to set access rules. After its first release, revisit the workload profile, since new AI features and reports often shift which tables deserve denormalisation.